

**PROJECT REPORT**

**SUBMITTED BY:**

***HANZLA SIBGHAT (19i-0515)***

***Muhammad Fozan (19i-0507)***

***MUHAMMAD BILAL (19i-0636)***

**SECTION:**

***D***

**Introduction:**

We created an AI for the Car racing game Torcs in this project. We were given a server that hosted the race, and we connected to it via the client, then manually drove the car using the arrow keys.

**Outputs:**

Multiple automobile directives were received from the game server. We utilized the following to implement the AI:

* Speed
* Gear
* Distance from edge (from multiple sensors)
* Time
* Opponents
* Angle with the track axis
* Distance from the track axis

**Inputs:**

The inputs our AI sends to the game server were:

* Acceleration
* Gear
* Acceleration
* Brakes
* Steering

**Logic:**

Our AI made decisions based on a variety of scenarios, which are detailed below:

* If the car is travelling in a straight line, it should accelerate. This implies that the acceleration value should be high (closer to 1).
* The car should turn if a turn is near. This implies that the steering value should be adjusted so that the car turns toward the track axis (angle should be closer to 0). If a sharp turn is approaching then the car should slow down along with turning. This means that the value of acceleration should decrease and the value of brakes should increase along with the change in steering value.
* From the stored dataset, we will filter the least required features and use only those contributing to the action and then train a model accordingly.

**Model:**

We are using Neural Networks to predict the output given the values of sensors. We used neural networks because it was giving minimum loss i.e.,0.02 approx. with higher accuracy of 90 approx. The model predicts accurate values but has a little higher ping for prediction which affects results. In order to compensate the loss, we used average of 15 rows from the dataset to form another dataset and then used it to construct the model. The Layers Used are Input, Dense and Dropout. The designed model is Shared Layer model with each branch training against each activator like accel, brake and steer. The activation functions are linear because steering was negative so using Relu was not useful in this case and was affecting the accuracy.

**Conclusion:**

The AI performs good on straight tracks and minor turns but whenever a sharp turn approaches it moves a little too much or too late. The main reason behind maximum wrong control is prediction ms due to which action keeps on repeating until new prediction is made which takes time like 100-250 ms. Whereas, the response time for Torcs server is expected to be 20 ms.